REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
31-03-2010	Final Report	10/19/2005 - 12/31/2009

4. TITLE AND SUBTITLE
Semantic Information eXchange Architecture (SIXA).

5a. CONTRACT NUMBER
N00014-05-C-0367

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Christopher J. Matheus
Mieczyslaw Kokar
Jakub Moskal
Geoff Hanson

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
VIStology, Inc. 5 Mountainview Drive, Framingham, MA 01701
Northeastern University, 360 Huntington Avenue, Boston, MA 02115
Referentia Systems, Inc. 550 Paiea Street, Suite 236, Honolulu, HI 96819

8. PERFORMING ORGANIZATION REPORT NUMBER
SIXA Final

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Gary Toth
Office of Naval Research
875 North Randolph Street, Suite 1425
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES
Work involved subcontractors; Notheastern University and Referentia Systems, Inc.

14. ABSTRACT

The Semantic Information eXchange Architecture Project (SIXA) funded joint effort between VIStology, Inc., Referentia Systems, Inc. and Northeastern University. The main accomplishments of this work include the:
1. Design and partial development of the SIXA Semantic Web Service.
2. Development of translation scripts to transform JC3IEDM into OWL ontology.
3. Development of suspicious activity scenario, ontology and supporting simulated data.
4. Integration of SIXA with XCOP and the demonstration of its use in detecting suspicious activity.
5. Design and development of the SIXA AIS Processing Platform (SAPP).
6. Design and development of the LiveKB framework for the intelligent control of devices such as trackers.
These primary accomplishments are summarized in this final report.

15. SUBJECT TERMS
battlespace information, data mediation, information interchange, semantic web technologies, ontological reasoning, data pedigree, JC3IEDM, CNDE, suspicious activity.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Christopher J. Matheus
U	U	U	UU	17	19b. TELEPHONE NUMBER *(Include area code)* 508 - 788 - 5088

Reset

Phase II Final Report:
Semantic Information eXchange Architecture (SIXA)

VIStology, Inc.
Northeastern University
Referentia Systems, Inc.

Christopher J. Matheus, PI
Mieczyslaw M. Kokar
Jakub Moskal
Geoff Hanson

Introduction

The SIXA (Semantic Information eXchange Architecture) Project was an STTR funded joint effort between VIStology Inc., Referentia Inc. and Northeastern University that spanned the years of 2006-2009. The main accomplishments of this work include the

- Design and partial development of the SIXA Semantic Web Service
- Development of translation scripts to transform JC3IEDM into an OWL ontology
- Development of a suspicious activity scenario, ontology and supporting simulated data
- Integration of SIXA with XCOP and the demonstration of its use in detecting suspicious activity
- Design and development of the SIXA AIS Processing Platform (SAPP)
- Design and development of the LiveKB framework for the intelligent control of devices such as trackers (Ph.D. dissertation)
- Publication of five papers

These primary accomplishments are summarized in the rest of this final report.

SIXA Semantic Web Services Architecture

The initial architecture design of the SIXA Semantic Web Service, shown in Figure 1, consisted of three major layers:

- Data Sources - servers, web services and databases that provide track data for SIXA server. Optimally this data is stored as XML and contains Automatic Identification System (AIS), OTH-GOLD or (as it would eventually turn out) CNDE track data.

- Business Logic - this is where the source data is pulled, merged and processed. It exposes its functionality to the clients via a web service.

- Clients - applications that use the Track Data web service. These can be standalone applications or web applications that are accessed via a web browser.

The SIXA server can fetch source data in multiple ways including exchanging SOAP or REST messages with web services and establishing HTTP/HTTPS, RPC or JMS connections with appropriate servers. The track data would be translated using Mediation ontology into a common core ontology, which initially was to be JC3IEDM but eventually turned out to be CNDE. A Pedigree ontology (developed in Phase I) would augment the core ontology by capturing meta information about the source, collection conditions and processing of the track data. Inference rules would be used to analyze the track data and its pedigree information to make decisions about the credibility and reliability of the data. The results would be passed on to the end user through web-based clients, which would also allow users to submit semantically-rich queries to the system.

We implemented aspects of this initial design including JMS services, the JC3IEDM OWL ontology (see below), the JC3IEDM mediation translator, OWL-S service descriptions (to describe the SIXA services), a simple SVG viewer of the track data for the clients and sample scenarios and data sets (see below). Once the decision was made to focus on the integration with XCOP/CNDE (see below) several aspects of the initial design were altered (e.g. JC3IEDM swapped out for CNDE) or put on hold (e.g. AIS and OTH-GOLD support).

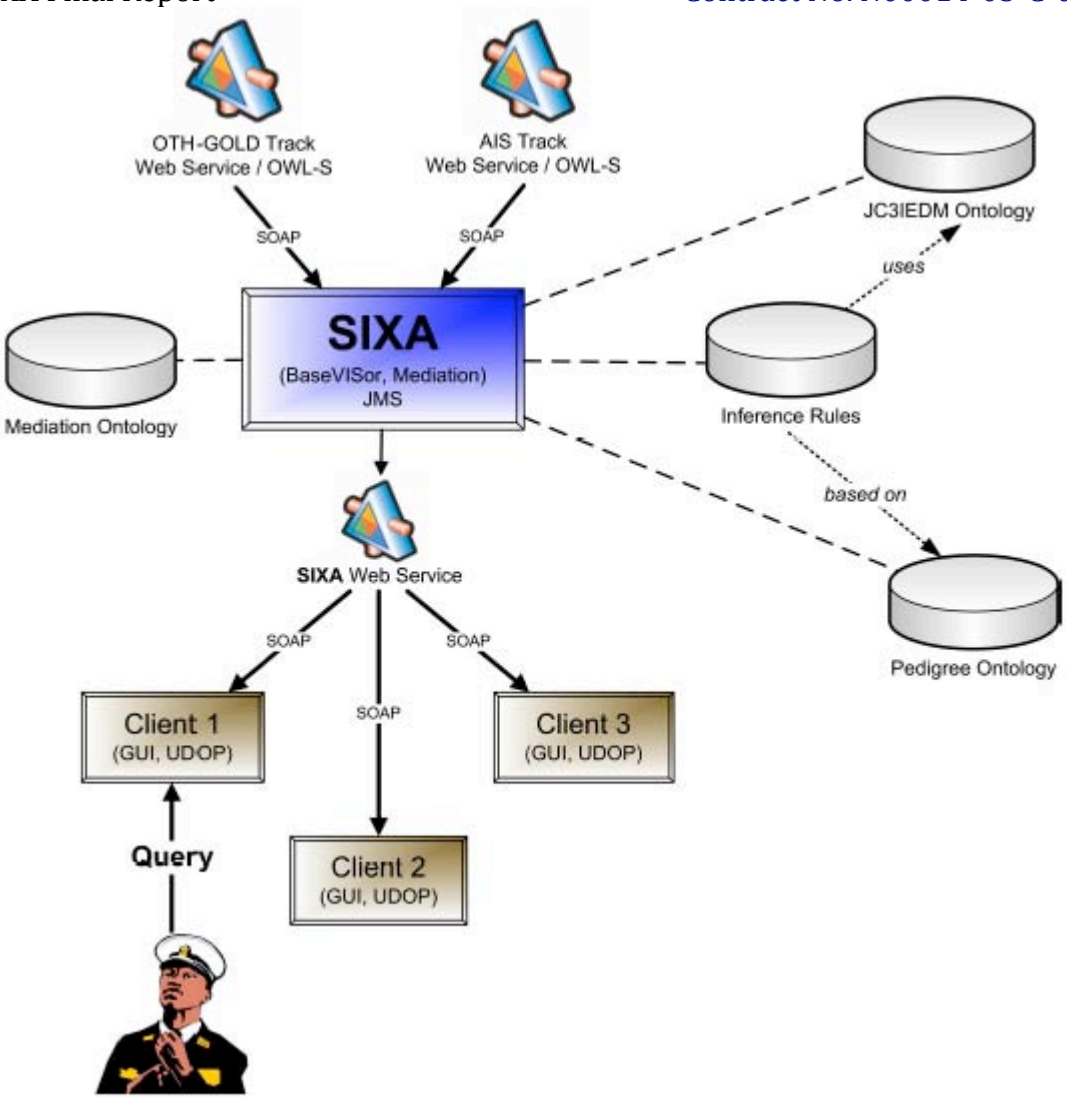

Figure 1. The initial SIXA Architecture

JC3IEDM Ontology

In Phase I we developed a translation of C2IEDM to OWL so that we could use it for representing track data along with its pedigree information. In Phase II we updated our translation scripts to do the same for JC3IEDM. We have made this ontology available for free on our web site at http://www.vistology.com/onts/onts.html and we published a paper about it at ICCRTS (see Publications section below). We had planned to use this translation as the primary data ontology in SIXA but later changed over to the use of CNDE in order to integrate more closely with XCOP.

Suspicious Activity Scenario and Ontology

To test the capabilities of SIXA we selected the problem domain of identifying suspicious activity in maritime vessels. Towards this end we developed a suspicious

activity OWL ontology (see Figure 2) and developed two scenarios with supporting simulated data.

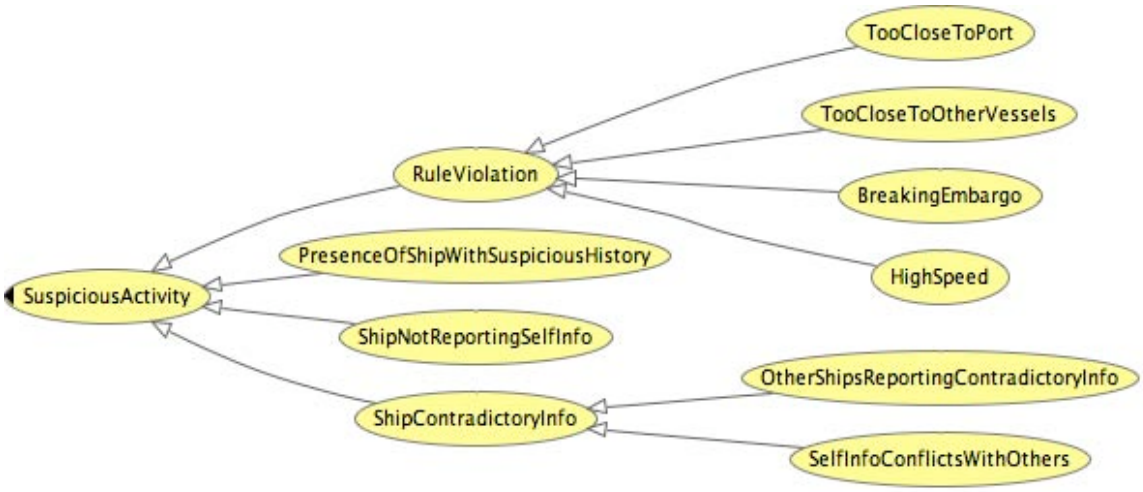

Figure 2 Suspicious Activity Taxonomy

In scenario 1 (see Figure 3) a USS Cole-style threat is developing in which the Blue ship is the target, berthed at a port when the Red ship approaches through harbor traffic and comes along side the Blue ship just before setting off a detonation; information about the approach Red ship is received from multiple sources with varying degrees of reliable pedigree.

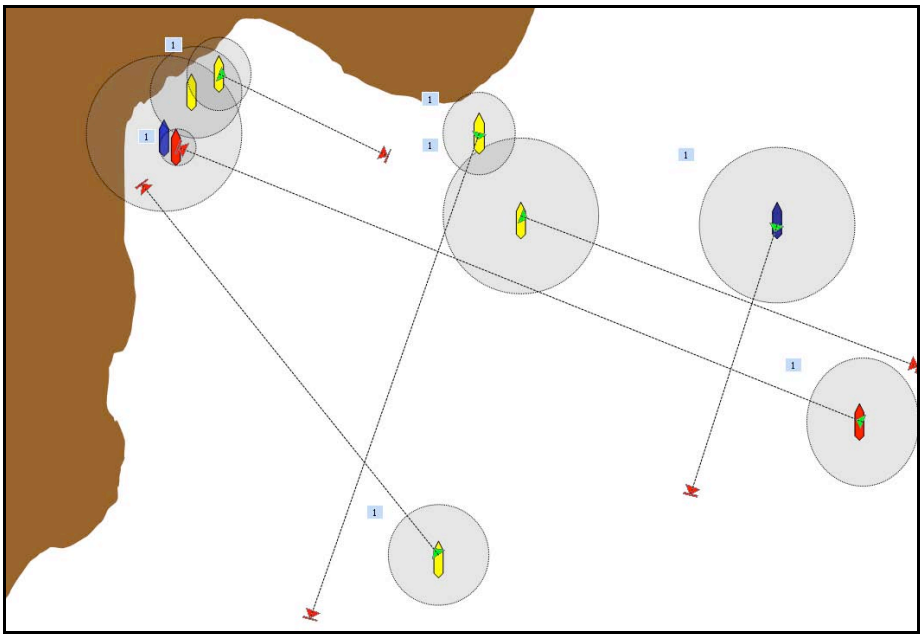

Figure 3 Suspicious Activity Scenario 1

In scenario 2 (see Figure 4) there is no Blue ship, only a Red ship which is breaking an embargo and as it does so it broadcast false AIS data about itself.

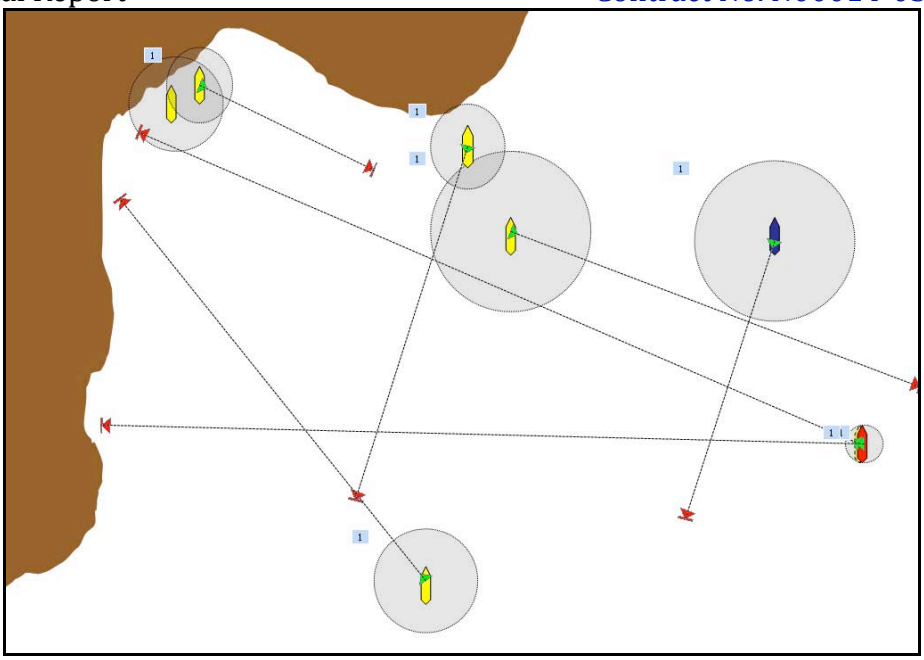

Figure 4. Suspicious Activity Scenario 2

While we generated data for the first of these two scenarios using JSAF, the project took a turn (integration with XCOP) before we got to the point of implementing the reasoner and processing the data.

XCOP Integration

Early in Phase II we identified XCOP as a possible vehicle for demonstrating the concepts being put forth in SIXA. In the spring of 2007 we were given the opportunity to work with the XCOP team to test some of the SIXA pedigree reasoning capabilities through integration with XCOP. The first step was to switch from JC3IEDM to the CNDE data model. We developed a CNDE OWL ontology along with scripts from translated raw CNDE data in OWL annotations.

Next we worked with XCOP and SPAWAR personnel and SMEs to develop a scenario to test the use of pedigree information in helping in the analysis of suspicious maritime activities. The specific scenario included two vessels, A and B, three sources of vessel tracking information, C, D, and E, and a correlator that fuses tracks from track sources C and E. Track source C is on an island in a harbor and is the target of a compromising attack by vessel B which intends to force C to produce false information about the location of its accomplish ship A. Source C has a posted "exclusion zone" that prohbits vessels from being within X meters of it.
A SIXA reasoner is also in the picture. It has pedigree information indicating that 1) the Correlator is processing data from C and E, 2) the Correlator and Sensor D are highly reliable, 3) the relaibility of A and B is unknown or less than reliable nd 4) C and E are reliable.

The sceanrio is broken down into four time steps, T1-T4 (see Figure 5 through Figure 8). At time T1 both ships are headed towards C, B coming from the east and A from the south:

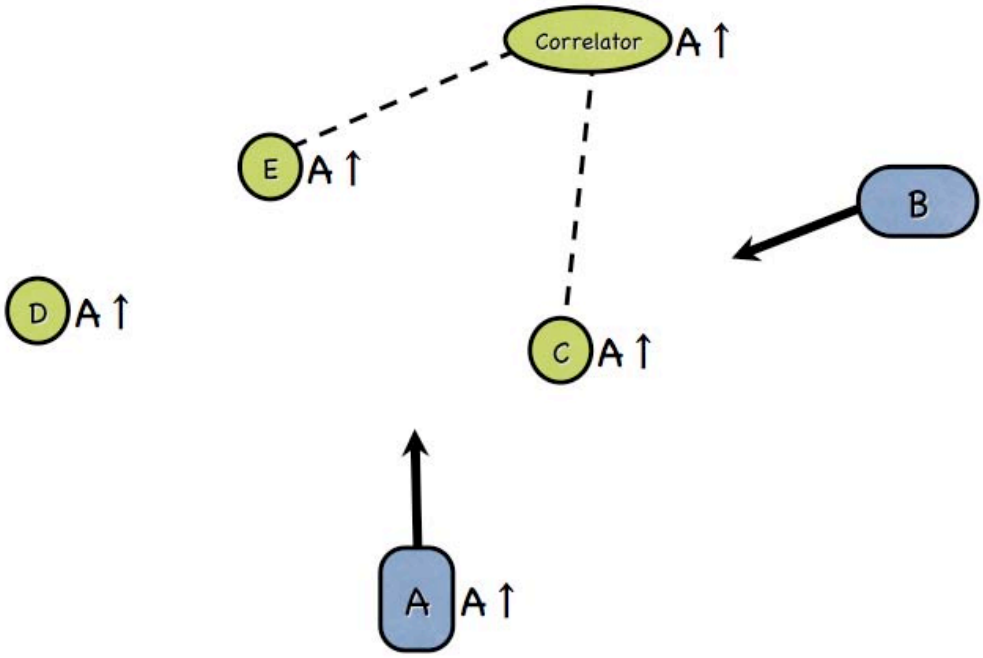

Figure 5. SIXA-XCOP Scenario: at T1. The up arrows next to each track source represents the direction it is reporting that ship A is headed.

At time T2 ship B is at positioned at the location of C, violating its exclusion zone, and is in the process of compromising C's track feed while A continues on its northward track:

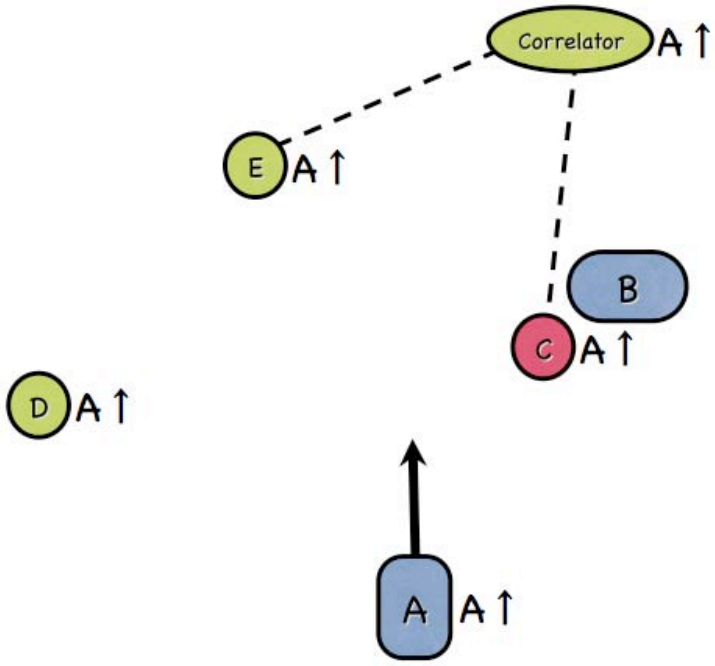

Figure 6. SIXA-XCOP Scenario at T2. Ship B is in the exclusion zone of C and is attempting to take over control of its track reporting.

At time T3 B has successfully taken over control of C's reporting and begins to move away.

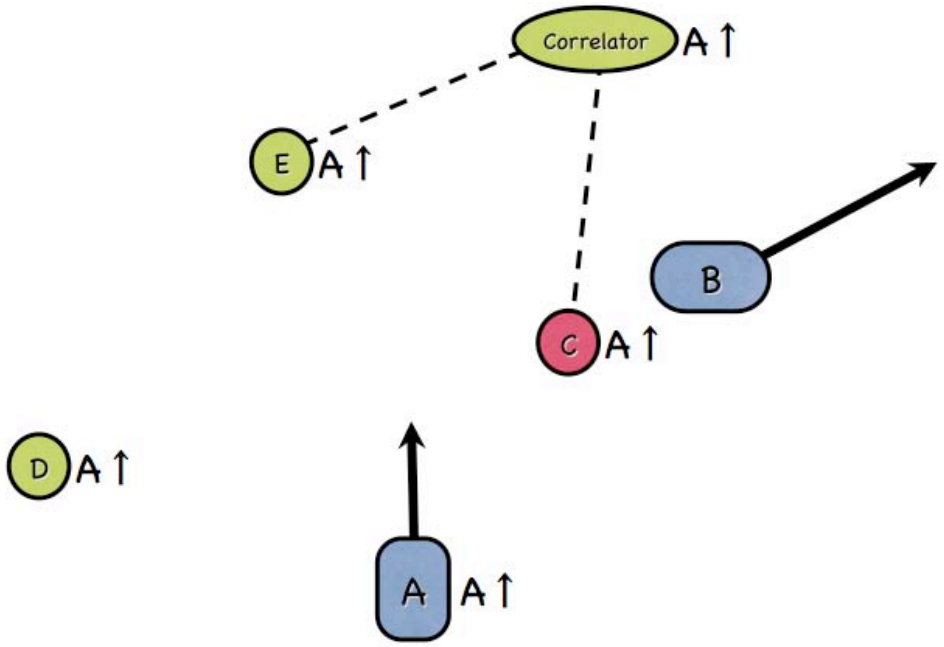

Figure 7. SIXA-XCOP Scenario at T3

At time T4 B is well away from C while still being in control of its feed and causes it to begin sending out false information about A's location and heading. At the same time A begins sending out false AIS information about its location and heading in accordance with C's false information. D and E continue to deliver accurate information about A but the correlator is now receiving contradictory information about A from E and C.

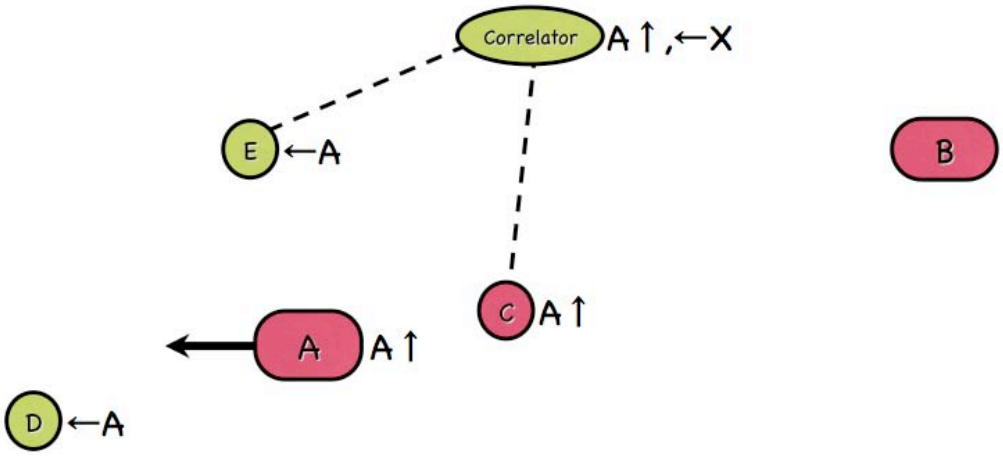

Figure 8. SIXA-XCOP Scenario at T4. Ship A is headed west but it and source C are reporting it as heading north.

The objective for SIXA wss to be able to identify that C has possibly been compromised and when contradictory data is being received form C and E to produce an alert to the fact of the compromise and identify C as being suspect.

A Java program was developed to generate data necessary for this scenario. It took as input the high level paths of ships A and B and generated XCOP track reports at regular intervals which where then fed into XCOP as shown in Figure 9. This figure also shows how the track information in XCOP flowed to both Google Earth for visual display and to SIXA for processing. In this setup, SIXA periodically queried XCOP using xquery for the status of all vessels in the area of operation. The requested data was retuned to SIXA a serialized CNDE mark up which SIXA converted into OWL before sending it to the BaseVISor inference engine for processing. When a suspicious activity regarding exclusion zone violations were detected or a decision concerning contradictory track information was need, SIXA would generate a KML message for display as an overlay on Google Earth.

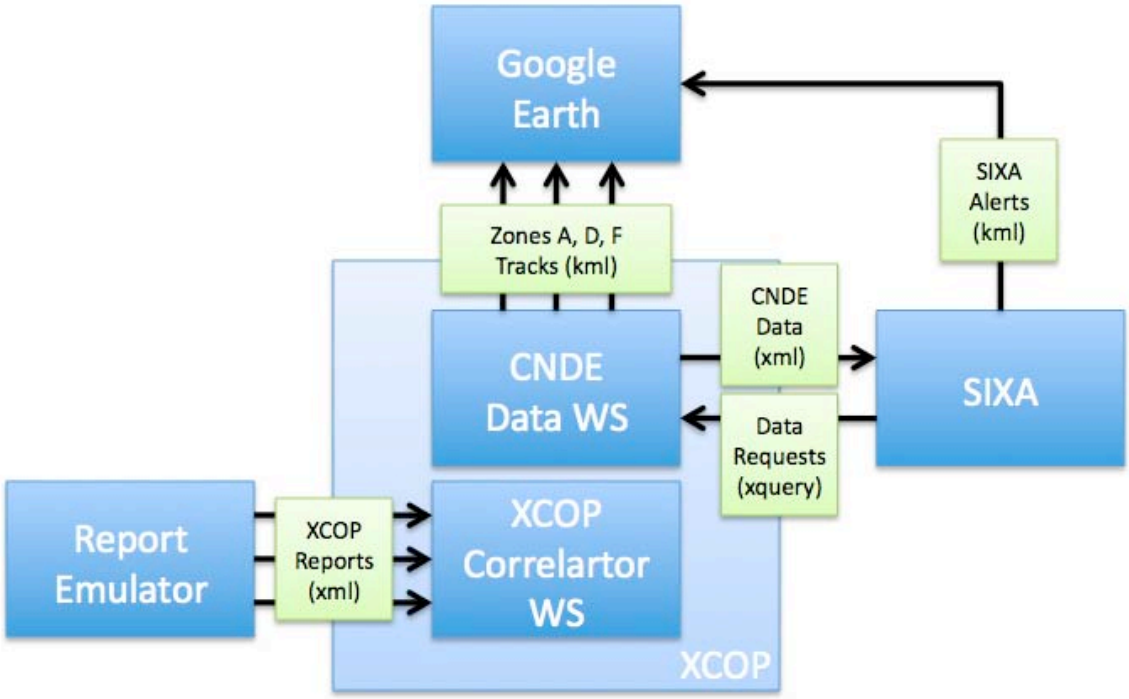

Figure 9. SIXA-XCOP Experiment Workflow

The SIXA-XCOP integration was successfully demonstrated at SPAWAR San Diego in December of 2007. As required, SIXA effectively integrated with XCOP, translated CNDE into OWL, reasoned about the information over the duration of the scenario and alerted via Google Earth notifications the exclusion zone violation and the likely compromising of track source C.

SAPP

We developed the SIXA AIS Processing Platform (SAPP) to enable the application of SIXA capabilities to streaming AIS data. As shown in the Component Architecture in

Figure 10, the system connects to an AIS web service, decodes the incoming AIS messages and constructs object models for each vessel using the ontology shown in Figure 11. SAPP subsequently records this information (both vessel information and position report) in the form of RDF/OWL triples to a MySQL database and generates KML messages for viewing the vessels on Google Earth in real time. The BaseVISor inference engine is coupled either with the database or directly to the vessel object manager (not shown in the diagram). Upon each update of a vessel's position information, a property is added to the information called isMostRecent (see the ontology) and this property is removed from the previous most recent information. Using this information, BaseVISor can efficiently query the database for the most recent information on all vessels giving an effective snapshot in time while still providing access to historical information.

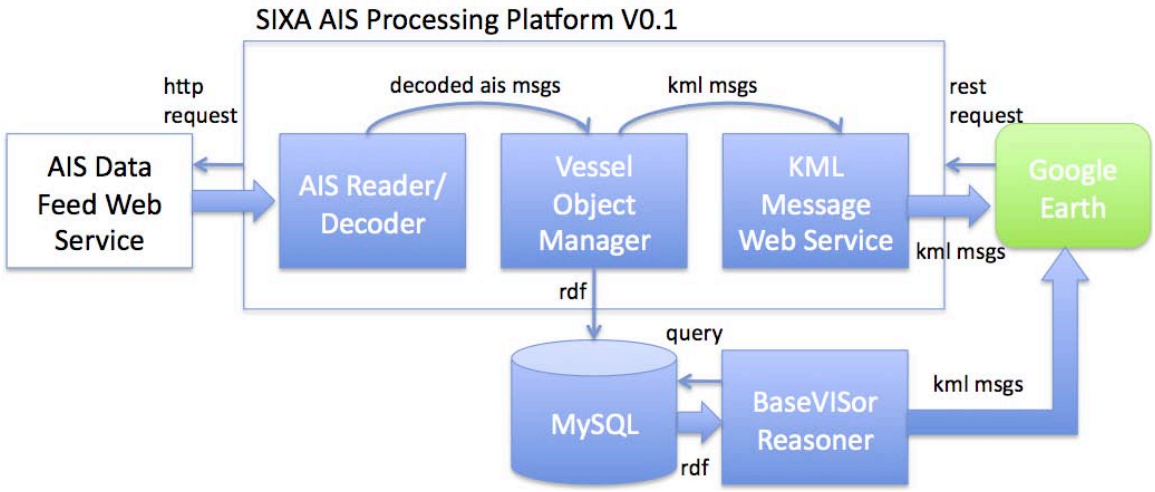

Figure 10 SAPP Component Architecture and Dataflow

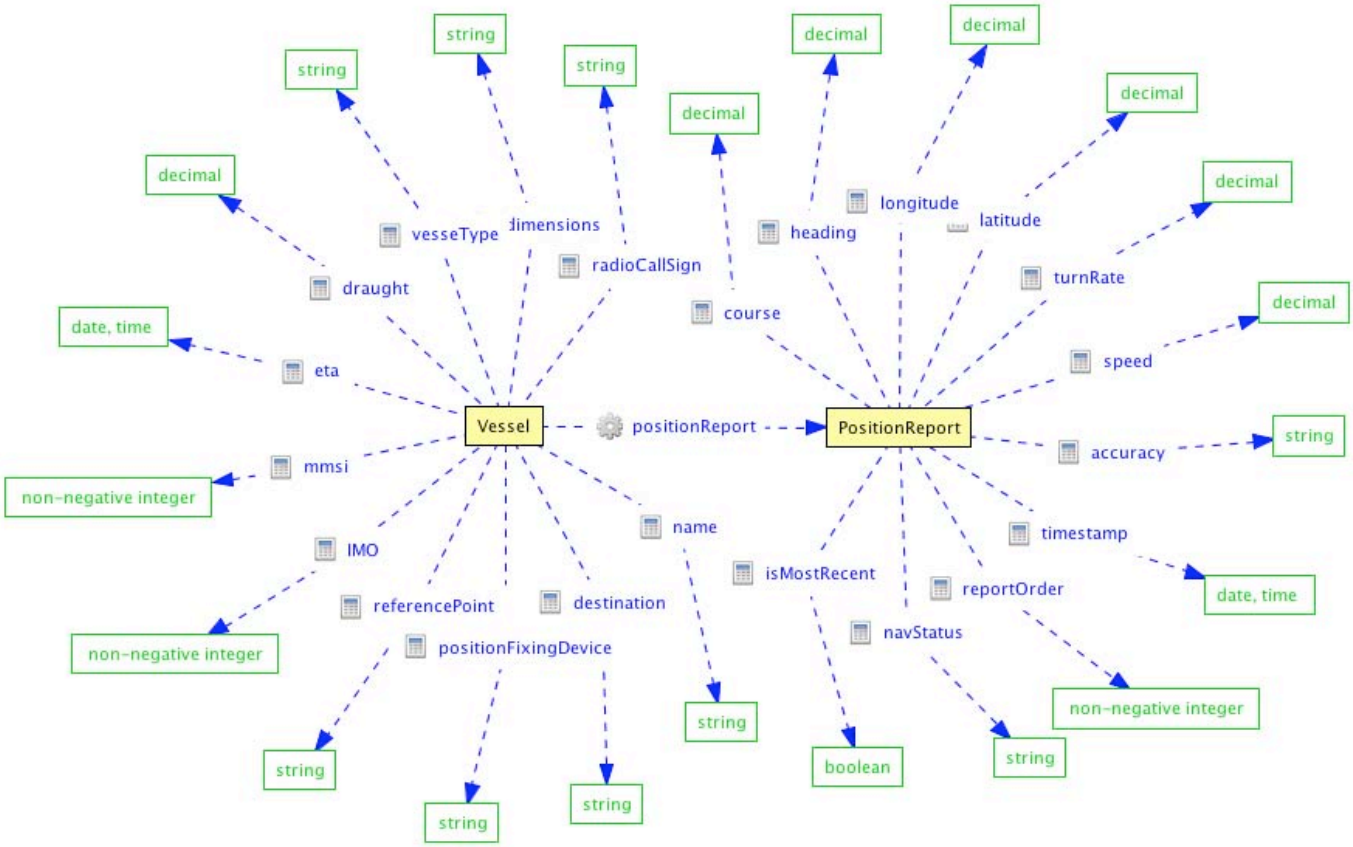

Figure 11 SAPP AIS Ontology

The system as designed was fully implemented and was tested with some simple rules written to analyze snapshots. While we did not have the resources to implement more sophisticated rules for suspicious activities nor to integrate the simple tracker we developed as we had hoped, the system has been reused for demonstration purposes in our subsequently ONR funded work under UPSIDE and will likely be used elsewhere.

LiveKB Framework

LiveKB is a framework that allows a knowledge-based approach to controlling devices and applications based on incoming contextual information. It is the focus of the disertation of Jakub Moskal whose gradutate assistanceship at Northereastern was funded through SIXA.

The first application that this approach was applied to was a simple tracking software algorithm that has parameters that are observable and adjustable from the outside. Coupling LiveKB to this application results in an enhanced tracker with improved performance.

Figure 1 illustrates the architecture of the enhanced tracker framework. The tracker receives radar observations from the sensors, fuses them together and produces

tracks which may be further processed or visualized using additional software. Our goal was to monitor and control the tracking software performance without requiring major changes to the existing components. The reasoner monitors the tracker and based on the provided rules, adjusts its parameters in order to satisfy some goals.

In order to allow the framework to be widely adopted it has to satisfy three principal requirements – *portability*, *flexibility* and *platform independence*. In order to meet these requirements we developed a LiveKB component, which mediates the interaction between the tracking software and the reasoner. Figure 13 shows the framework enhanced with the LiveKB. The main innovation here is that instead of interacting with the tracking software using some API, thus hard-coding the interaction, LiveKB uses information that is provided to it dynamically. The API is represented as a model, which is also mapped to the tracking ontology. Both the model and the mapping are provided to the LiveKB, which by the means of reflection and the CORBA middleware interfaces with the tracker in a platform independent manner. The Reasoner retrieves the context information and updates the tracker's parameters using the LiveKB component. This interaction is generic and not part of the implementation itself; the reasoner uses only ontological terms, which are translated to the proper method invocations using the provided mapping between the ontology and the tracker's model. On an update request from the reasoner, the LiveKB updates the parameters in the tracking software and also the ontology in order to reflect these changes in the reasoner's Knowledge Base (KB).

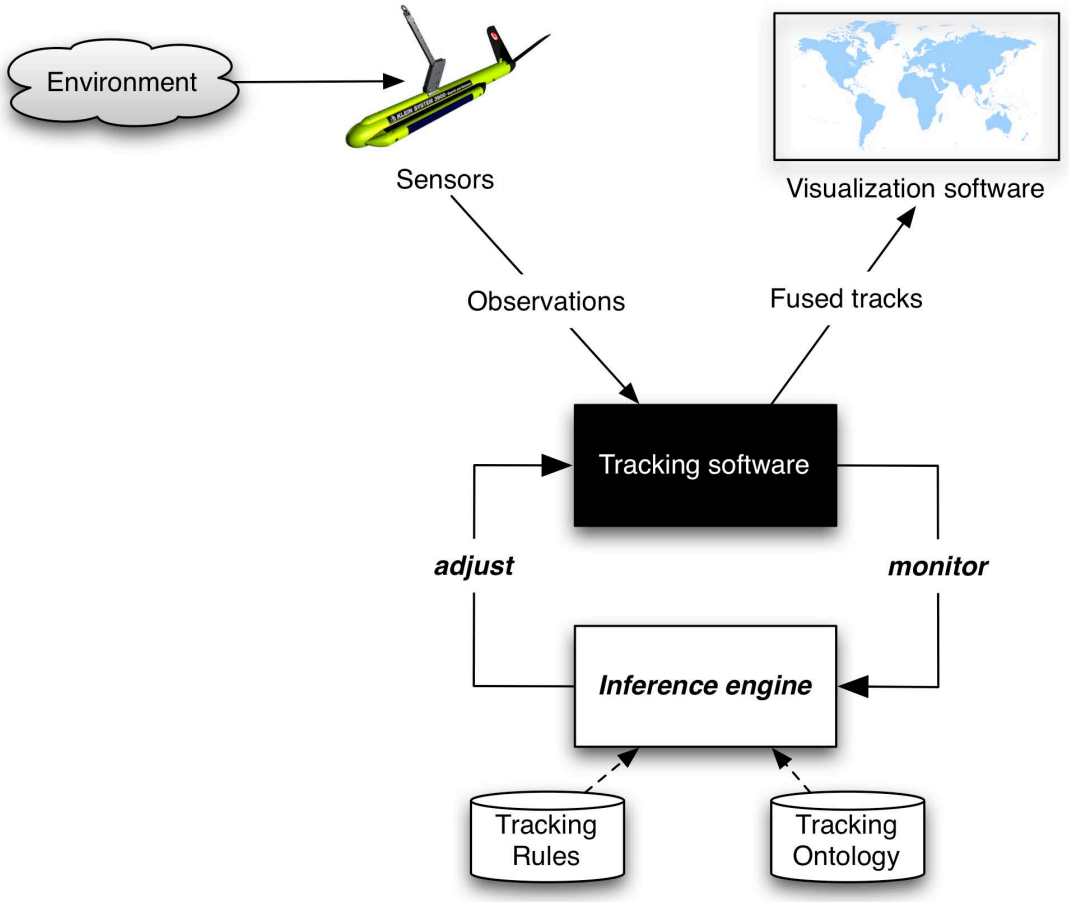

Figure 12 LiveKB Architecture for an Enhanced Tracker.

The framework enhanced with the LiveKB component satisfies all of our requirements:

- Portability – rules and the ontology are independent of the tracking software. The ontology may become an industry standard and be utilized in other scenarios, such as knowledge sharing between different trackers.
- Flexibility – the framework is not bound to a particular API because the hard-coded interaction is generic. All platform specific information is provided dynamically, thus it can be easily changed.
- Platform-independence – this is achieved due the use of CORBA middleware, a well-developed industry standard with the support for most programming languages.

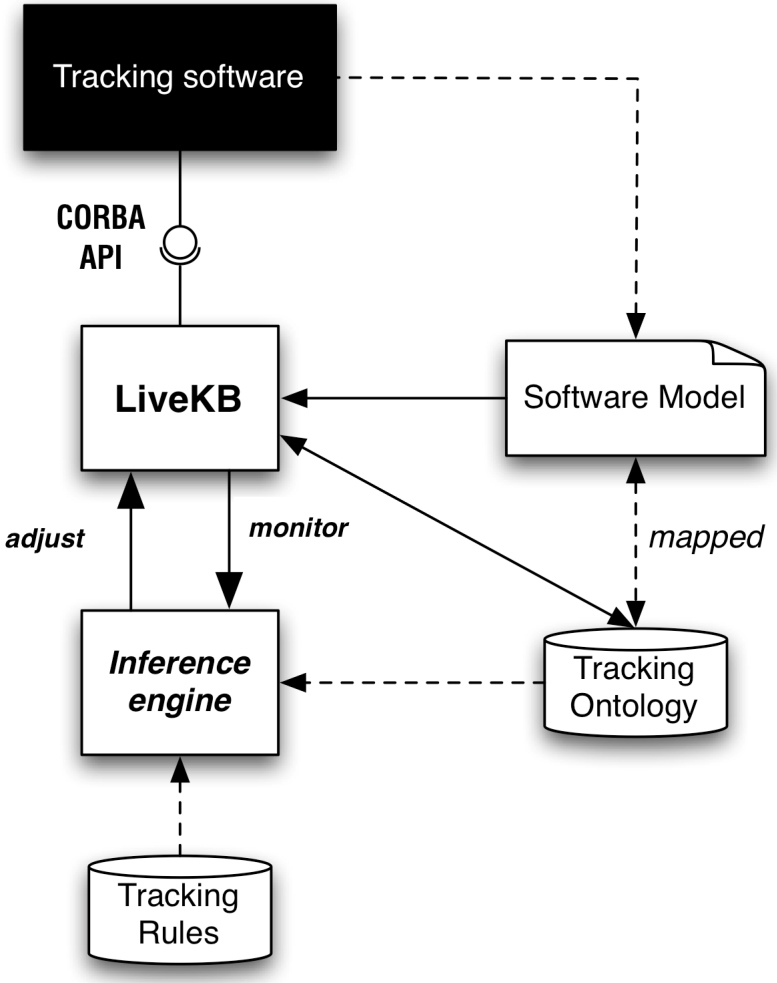

Figure 13: Tracking software interfaced with the reasoner using the LiveKB component

A prototype of LiveKB has been implemented and used to interface a BaseVISor reasoner with a Kalman Filter simulator (shown in Figure 14). The environment changes were simulated by manual parameter controls, which were then translated by the LiveKB into ontological terms and processed by the BaseVISor rules. Simple rules detected changes and reacted by updating other parameters in the Kalman Filter, reflected both in the Kalman software itself and in the BaseVISor's Knowledge Base. BaseVISor, not shown in the Figure 14, was periodically running in the background and checking the parameter values using the LiveKB.

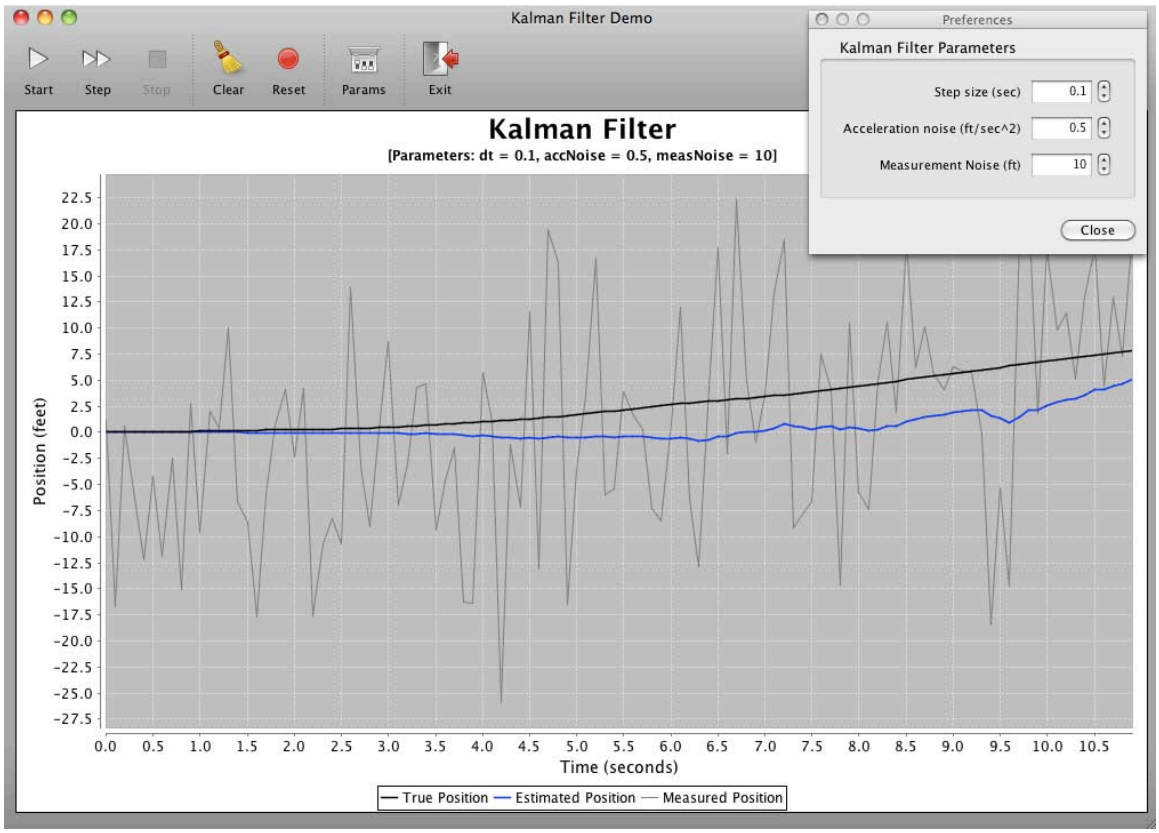

Figure 14: Kalman Filter simulator with adjustable parameters to simulate environment changes

Publications

The work performed in SIXA resulted in or contributed to the publication of five papers. PDF versions of these papers are available from our web site at http://www.vistology.com/papers.

This first paper describes the use of the LiveKB framework (initially developed to control a tracker) to control the interaction between a reasoner (BaseVISor) and Software Defined Radios. This paper won an award for Best Student Paper.

> J. Moskal and M. Kokar, Interfacing a Reasoner with an SDR: a Platform and Domain API Independent Approach Presentation. In Proceedings of the SDR '09 Technical Conference and Product Exposition, Washington, D.C., December 2009.

In this second paper SIXA and the rules developed to identify suspicious activity were used as the basis for comparing the usability of various rule engines.

> J. Moskal and C. Matheus, Detection of Suspicious Activity Using Different Rule Engines: Comparison of BaseVISor, Jena and Jess Rule Engines, In Proc. of The International RuleML Symposium on Rule Interchange and Applications, RuleML08, Orlando, FL, October 2008.

The third paper is a follow-on to an earlier paper written in Phase I on converting C2IEDM to OWL.

> C. Matheus and B. Ulicny, On the Automatic Generation of an OWL Ontology based on the Joint C3 Information Exchange Data Model. 12th International Command and Control Research and Technology Symposium, Newport, RI, June 2007.

These two papers describe BaseVISor, VIStology's inference engine optimized for processing OWL and RDF, including some of the extensions that were added to it under the SIXA project.

> C. Matheus, K. Baclawski and M. Kokar. BaseVISor: A Triples-Based Inference Engine Outfitted to Process RuleML and R-Entailment Rules. In Proceedings of the 2nd International Conference on Rules and Rule Languages for the Semantic Web, Athens, GA, November 2006.

> C. Matheus, B. Dionne, D. Parent, K. Baclawski and M. Kokar. BaseVISor: A Forward-Chaining Inference Engine Optimized for RDF/OWL Triples. In Digital Proceedings of the 5th International Semantic Web Conference, ISWC 2006, Athens, GA, November 2006.

Conclusions

The SIXA R&D effort progressed through a number of stages resulting in the development of several reusable capabilities including most significantly the translation of JC3IEDM into OWL, the core SIXA reasoning capability of processing CNDE track data and searching out patterns of maritime activities declaratively described using rules, the SAPP platform for processing streaming AIS data and applying rules to them and the LiveKB framework for controlling devices and applications based on incoming contextual information. Much of this groundwork is continuing to be used or further developed at VISTology and Northeastern: development of LiveKB is continuing in the form of a Ph.D. dissertation at Northereastern; the pedigree ontology developed in SIXA served as the basis for the Provenence ontology being developed for the ONR FTN-0904 UPSIDE project; the SAPP is being used in both UPSIDE and the ONR directed DiSIS (Distributed Semantically-integrated Information Services) effort (a new SBIR grant to VIStology, Inc.); the contributions made in SIXA to the development and further extension of BaseVISor are being used by all on-going projects within VIStology, Inc. as well in some research efforts being conducted at Northerastern.